



Original software publication

OptiSens—Convex optimization of sensor and actuator placement for ultrasonic guided-wave based structural health monitoring

Sergio Cantero-Chinchilla^{a,d}, James L. Beck^b, Juan Chiachío^{c,e}, Manuel Chiachío^{c,e}, Dimitrios Chronopoulos^{a,*}^a Institute for Aerospace Technology & The Composites Group, The University of Nottingham, NG7 2RD, United Kingdom^b Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125, United States^c Department Structural Mechanics & Hydraulics Engineering, University of Granada, 18001, Spain^d Aernnova Engineering Division S.A., Madrid, 28034, Spain^e Andalusian Institute in Data Science and Computational Intelligence, Granada, Spain

ARTICLE INFO

Article history:

Received 23 September 2020

Received in revised form 2 December 2020

Accepted 3 December 2020

Keywords:

Optimal sensor placement
Structural health monitoring
Ultrasonic guided-waves
Cost-benefit optimization

ABSTRACT

This paper presents OptiSens, a computational platform in Python and Matlab, that provides optimal sensor and actuator configurations for structural health monitoring applications using ultrasonic guided-waves. This software formulates a convex entropy-based objective function, which aims at minimizing the uncertainty while maximizing the expected accuracy of the monitoring system in localizing structural damage. The platform is specialized for two types of different materials, namely isotropic and composite (anisotropic) materials. The effectiveness and efficiency of this software are demonstrated using two plate-like structures made of different materials.

© 2020 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Code metadata

| | |
|---|--|
| Current code version | Version 1.0 |
| Permanent link to code/repository used for this code version | https://github.com/ElsevierSoftwareX/SOFTX-D-20-00052 |
| Code Ocean compute capsule | none |
| Legal Code License | MIT License |
| Code versioning system used | none |
| Software code languages, tools, and services used | Python and MATLAB. |
| Compilation requirements, operating environments & dependencies | The Python package requires Python, numpy, scipy, matplotlib, and pathlib. The MATLAB package requires MATLAB 2017b and above. Documentation provided within code. |
| If available Link to developer documentation/manual | scanteroch@gmail.com |
| Support email for questions | |

1. Motivation and significance

Structural health monitoring (SHM) deals with the development of technologies for the online and automated detection, localization, classification and identification of damage in engineering structures [1–3]. Sensing is the cornerstone of any functional SHM technology, with the number of sensors and their position being a decision of upmost importance. The use of redundant sensors in any SHM system has potential safety-related

benefits to ensure the resilience towards malfunctioning and data contamination. However, employing more sensors than the strictly needed increases not only the cost, weight and installation complexity, but also the measurement errors. Thus, determining the optimal number of sensors along with their optimal positioning on the structural ensemble is of paramount importance for an efficient and reliable SHM system.

The accuracy of the SHM systems also relies on the adopted technique to interrogate the structural health. Among the most well-known approaches, it is worth highlighting the use of vibration analysis [4,5] and fiber Bragg grating [6]. These SHM methods stand out due to their efficiency in monitoring very large structures such as buildings or bridges, although they generally lack of

* Corresponding author.

E-mail address: Dimitrios.Chronopoulos@nottingham.ac.uk (D. Chronopoulos).

sensitivity to small and local damage. In this sense, a promising technique is the use of ultrasonic guided-waves [7]. These are mechanical waves that have the property to travel and explore very large areas in thin-walled structural components with a relatively small attenuation [8]. Moreover, guided-waves are particularly sensitive to small damage forms in the early stage of growth, such as the onset of a fatigue crack. These remarkable characteristics have led industries, such as the aerospace industry, to focus on this diagnosis technology. One noticeable example of this development is the PAMELA[®] SHM system developed by Aernnova Engineering S.A. [9,10], consisting of an integrated hardware and software system, including sensors and actuators, able to perform ultrasonic guided-wave tests in thin-walled structures.

A number of researchers have worked on sensor placement optimization during the last decade. For example, Papatimitriou [11] proposed an optimal sensor placement methodology consisting of minimizing the entropy of the posterior distribution in vibration-based applications, which was further developed to account for spatial correlation of sensors in [12]. A Bayesian approach to minimize false positives and false negatives with application to guided waves was proposed in [13]. In addition, other authors have addressed the problem of optimal sensor placement for ultrasonic piezoelectric transducers using performance indices such as the probability of detection [14]; the area of coverage [15,16]; and the value of information [17]. As a result, a number of software platforms have been proposed in the literature. For instance, the software SPEM [18] was developed to optimize the sensor placement by maximizing the linear independence of vibration modes by the Fisher information matrix in civil engineering applications. In the same application field, the OSPS Matlab toolbox [19] uses the modal assurance criterion for the optimization of accelerometers. Another example, in the area of ultrasonic guided-wave based SHM, is SenOpT [20], which is a Matlab graphical user interface for the optimization of the number and location of piezoelectric sensors by maximizing the coverage area. However, there is no open source software tools for determining the optimal number and positioning of ultrasound-based SHM actuators and sensors for plate-like engineering structures from an information theoretic perspective.

This manuscript presents the OptiSens platform, both open source and semi-open source software interfaces developed in Python and Matlab, respectively, to test the optimal positioning of sensors and actuators in plate-like structures made of both isotropic and anisotropic materials. This platform provides answer to an increasing technological need of the SHM community, particularly among the research engineers and practitioners in the field of ultrasonic guided-wave based SHM. The scientific methodology behind OptiSens has been successfully published in [21] to tackle the configuration optimization for panels within an aerospace context; however, the methodology is generic and hence transferable to many civil, energy, naval, and mechanical infrastructure system with minor changes. The main objective of OptiSens is to become a benchmarking platform to compare against in terms of accuracy and efficiency.

OptiSens is designated so as to require minimal user-defined inputs for ease of use; nonetheless, several meta-parameters are required for its optimal performance, e.g. the geometry of the plate, the cost of sensors and actuators, or their possible locations in the plate. The resulting optimal sensor and actuator configurations are provided in different formats, including data files, figures, and tables. A full software description is provided below.

2. Software description

The current version of OptiSens enables the optimization of the number and position of piezoelectric sensors and actuators based on an entropy-based optimization criterion first presented in [21], which is concisely described here within the context of the software architecture under a unified notation.

2.1. Optimization problem

The optimization underlined in OptiSens is built upon the minimization of the Shannon's information entropy of a pre-posterior probability density function (PDF) that is based on anticipated data. Given that the design of any SHM system is addressed before it is physically built, the optimization relies on model-predicted (or anticipated) data, rather than experimental data. These data are used to obtain the information entropy of the pre-posterior (before the arrival of experimental data) PDF. Note that the entropy is a measure of the posterior parameter uncertainty. The simplified expression of the Shannon's entropy is given by:

$$h(w, z) = -\mathbb{E}_{\theta_0} [\log \det Q(w, z, \theta_0)] \quad (1)$$

where $w \in [0, 1] \subset \mathbb{R}$ and $z \in [0, 1] \subset \mathbb{R}$ are decision variables that control the presence or absence of an actuator and a sensor, respectively, θ_0 are uncertain parameters (e.g. damage coordinates or wave propagation velocity), and Q is the covariance matrix of a predefined stochastic model of ultrasonic time of flight, defined as:

$$Q = \sum_j^{N_a} w_j \sum_i^{N_s} z_i P_i^j(\theta_0) \quad (2)$$

where N_a and N_s are the specified maximum numbers of actuators and sensors, respectively. The matrix $P_i^j(\theta_0)$ is the Hessian matrix of the model evaluated in the i th sensor and the j th actuator. Such a model evaluation provides an estimation of the measurement acquired in the i th sensor when the structure is actuated by the j th actuator.

Eq. (1) only addresses the optimal placement of actuators and sensors for a specific number of them. Thus, to obtain the optimal number of sensors and actuators, a monotonically increasing dimensionless cost function $s(n)$ is considered to quantify the cost of adding n devices into the SHM system. The variable n is chosen to be a real number in the interval $n \in [0, (N_a + N_s)]$. Since the entropy-based objective function in Eq. (1) is monotonically decreasing with respect to n [11], and $s(n)$ is monotonically increasing, a convex minimization problem can be defined as follows:

$$\begin{aligned} &\underset{w, z, n}{\text{minimize}} && h^*(w, z, n) = -\mathbb{E}_{\theta_0} [\log \det Q(w, z, \theta_0)] + \eta \cdot s(n) \\ &\text{subject to} && 0 \leq w_j \leq 1, \quad j = 1, \dots, N_a \\ & && 0 \leq z_i \leq 1, \quad i = 1, \dots, N_s \\ & && \sum_{j=1}^{N_a} w_j + \sum_{i=1}^{N_s} z_i = n \end{aligned} \quad (3)$$

where $\eta > 0$ is used to establish a particular trade-off between information gain and cost. The function $s(n)$ is a dimensionless cost function formed using interpolating monotonic cubic splines [22], which takes a value of 0 when the amount of sensors and actuators are minimum, and 1 otherwise. As for the minimization in Eq. (3), standard convex minimization algorithms may be used; here the interior-point algorithm [23] and the sequential least squares programming [24] are used for both Matlab and Python implementations, respectively.

This convex optimization methodology is implemented to infer the damage position using an ellipse-based Bayesian model inverse problem [25] and the data from the time of flight (ToF) of the scattered wave from the damage to the sensors. To this end,

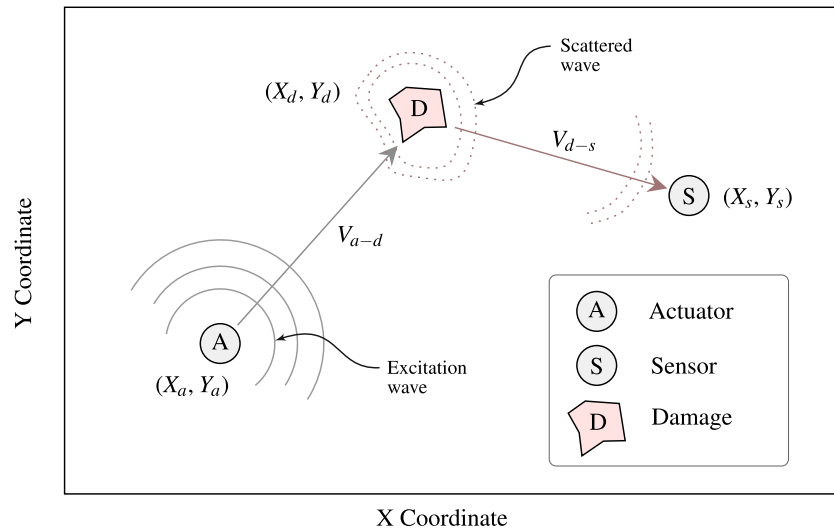


Fig. 1. Schematic of ultrasonic guided-wave based damage (D) localization using an actuator (A) and a sensor (S). The excitation guided-wave travels across the structure and interacts with D, which causes the appearance of a scattered wave from D. The sensor (S) measures the mechanical displacements carrying damage information, including its location.

the observations for each sensor–actuator pair are compared with the predictions of a ToF model [21], as follows:

$$x_i^j = \frac{\sqrt{(X_d - X_a^j)^2 + (Y_d - Y_a^j)^2}}{V_{a-d}(f, \alpha)} + \frac{\sqrt{(X_d - X_s^i)^2 + (Y_d - Y_s^i)^2}}{V_{d-s}(f, \alpha)} \quad (4)$$

where (X_d, Y_d) are the coordinates of the damage position, (X_a^j, Y_a^j) are the j th actuator transducer coordinates, (X_s^i, Y_s^i) are the coordinates of the i th sensor transducer, and $V_{a-d}(f, \alpha)$ and $V_{d-s}(f, \alpha)$ are the wave propagation velocities of the actuator–damage and damage–sensor paths defined by the angle α , respectively. Fig. 1 depicts the elements of Eq. (4) and the guided-wave propagation.

In general, the velocities depend on the frequency of excitation f , the wave mode under investigation (e.g., symmetric or anti-symmetric modes), and the direction of the path with respect to the material orientation. Assuming the more general case involving an orthotropic material, such as a composite laminate, these velocities can be approximated by modeling the velocity profile with respect to the angle for a given frequency. The velocity is modeled as an ellipse as in the case of angle-ply laminates, as follows [26]:

$$V(f, \alpha) = \sqrt{V_x^2 + V_y^2} = \sqrt{(a \cdot \cos(\gamma))^2 + (b \cdot \sin(\gamma))^2} \quad (5)$$

where the parameter angle γ relates to the physical angle α (formed by the paths between damage and actuator or sensor) [26] as follows:

$$\gamma = \arctan\left(\frac{a}{b} \tan(\alpha)\right); \quad \left\{ \alpha_a = \arctan\left(\frac{Y_d - Y_a^j}{X_d - X_a^j}\right), \quad \alpha_s = \arctan\left(\frac{Y_s^i - Y_d}{X_s^i - X_d}\right) \right\} \quad (6)$$

where α_a and α_s denote the geometrical angle formed by the actuator–damage and damage–sensor paths, respectively, and a and b denote the horizontal and vertical axes of the velocity ellipse. Therefore, the set of uncertain parameters of the ToF model for orthotropic materials consists of the damage coordinates along with a and b , so $\theta = \{X_d, Y_d, a, b\}$. Alternatively, for isotropic materials such as an aluminum plate, the set of parameters simplifies as $\theta = \{X_d, Y_d, V\}$.

3. Software architecture

OptiSens has been implemented in Python and Matlab programming languages so that a wider community can benefit by using this software. Nevertheless, both implementations have strong differences and are oriented to different types of users. While the Python version is oriented to developers and researchers who may want to continue developing the source code, the Matlab version is devoted to end-users, such as MSc/Ph.D. students and practitioners. The main architecture of the two implementations is described in this section.

3.1. Python code

The workflow of the optimization code for the Python implementation is shown in Fig. 2. The user starts by choosing the main file depending on the material of the plate-like structure: (1) OptiSensComp.py for composite materials which spatial distribution of the wave propagation velocity can be modeled by an ellipse; and (2) OptiSensMet.py for isotropic (e.g. metallic) plates.

Next, the user needs to enter input data about (1) the outer geometry of the plate by the coordinates of its vertices; (2) prior distributions of the model parameters described by an area of possible damage occurrence (in the form of damage coordinates) and the mean and standard deviation of the wave velocity variables; and (3) the distance and number of concentric rings for the grid of possible sensor and actuator locations (see Fig. 3). After running this module, two files are generated in the folder ./dat, which contain the samples from the prior distributions and the coordinates of the possible sensor and actuator locations. Note that the units of the coordinates are the same as the ones used for the input parameters.

Then, a database of P matrices (related to Q in Eq. (3) as shown in Eq. (2)) is obtained by evaluating the covariance matrix for each of the samples obtained from the prior distribution. This database is also saved in ./dat. Once the database is created, the optimization problem is run by defining the cost function $s(n)$ (a generic one is provided within the code) and an initial point where the minimization algorithm will start. Finally, the resulting optimal sensor and actuator configuration is stored in different files: Opt_Sol.npy containing the output of the optimization algorithm from Python, OptSol.pdf which provides a figure of

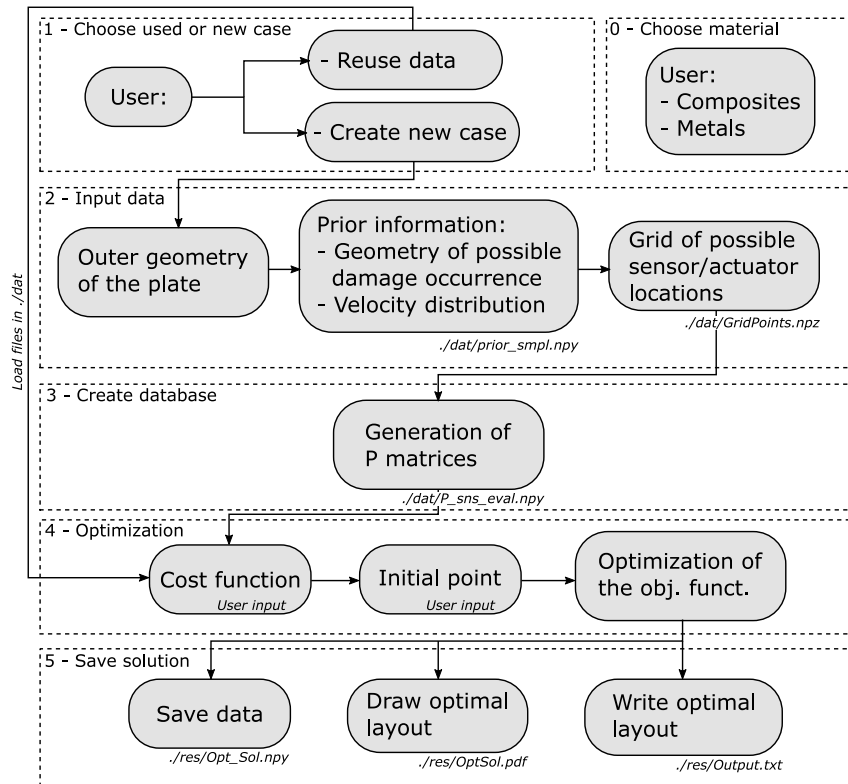


Fig. 2. Architecture of Python implementation of OptiSens.

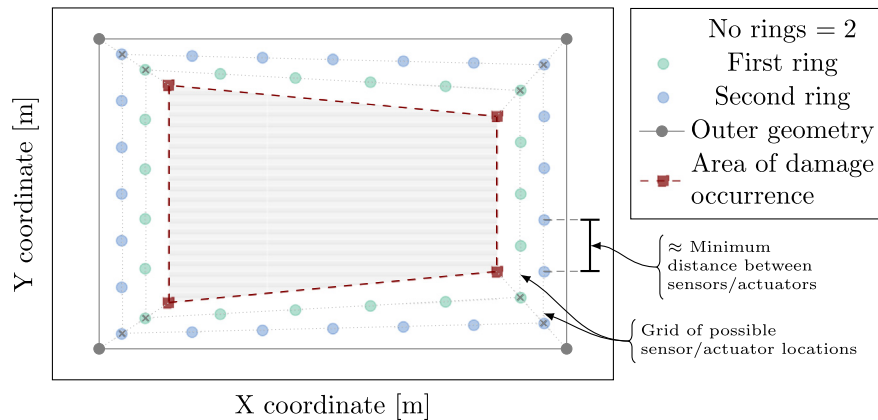


Fig. 3. Graphical representation of an example of the geometrical input variables.

the optimal configuration, and `Output.txt` containing a list with the best position of sensors and actuators along with the optimal number of sensors and actuators.

3.2. Matlab code

The Matlab implementation of OptiSens shares the main structure shown in Fig. 2. However, in this case, the input parameters are requested from the user by means of dialog boxes (e.g. see Fig. 4), which provides a more friendly environment for the designer, since there is no need to manipulate the code.

After choosing the material, the user selects the appropriate main Matlab file: (1) `OptiSensComp.m` for composite materials; and (2) `OptiSensMet.m` for isotropic plates. Then, a dialog box asks the user whether or not to reuse the input files in `./dat` folder from a previous case study, if any. In any case, several dialog boxes appear to introduce input data for the optimization

of sensors and actuators. Finally, three output files are produced and stored with analogous file names than those in the Python implementation.

3.3. Software functionalities

The main functions of OptiSens are shared by both the Python and Matlab implementations, which comprise the following features:

prior_smp1 Creates and saves a set of samples from the prior distribution of the model parameters θ_0 . Note that a uniform PDF is considered for the coordinates of the possible damage location (X_d, Y_d) . Note also that other distributions might be specified as appropriate.

gridPoints Generates the grid of possible sensor and actuator locations by specifying (1) a number of concentric rings

OptiSens - Outer geometry of the plate

HELP: Both arrays need to have the same length.

1. Enter an array of values for X coordinates in [m]:
[-1 , 1 , 1 , -0.9]

2. Enter an array of values for Y coordinates in [m]:
[-0.5, -0.5, -0.15, 0.5]

OK Cancel

OptiSens - Grid of possible sensor locations

HELP: These variables will create the grid in the space between the outer plate edges and the area of possible damage occurrence.

1. Enter the number of rings around the area of possible damage:
2

2. Enter the minimum distance between consecutive points in the same ring in [m]:
0.2

OK Cancel

OptiSens - Model parameters information

First, enter information related to X_d and Y_d :

HELP: Both arrays (1 and 2) need to have the same length between them, and the same number of vertices than the outer geometry.

1. Enter an array of values for X coordinates in [m]:
[-0.70, 0.70, 0.70, -0.65]

2. Enter an array of values for Y coordinates in [m]:
[-0.35, -0.35, -0.23, 0.22]

Second, enter wave propagation velocity information:

3. Enter the mean value of the velocity in [m/s]:
2800

4. Enter the standard deviation of the velocity:
60

Third, enter number of samples used for the objective function evaluation:

5. Enter number of samples:
10

OK Cancel

Fig. 4. Dialog boxes asking for input parameters in case of OptiSensMet.m.

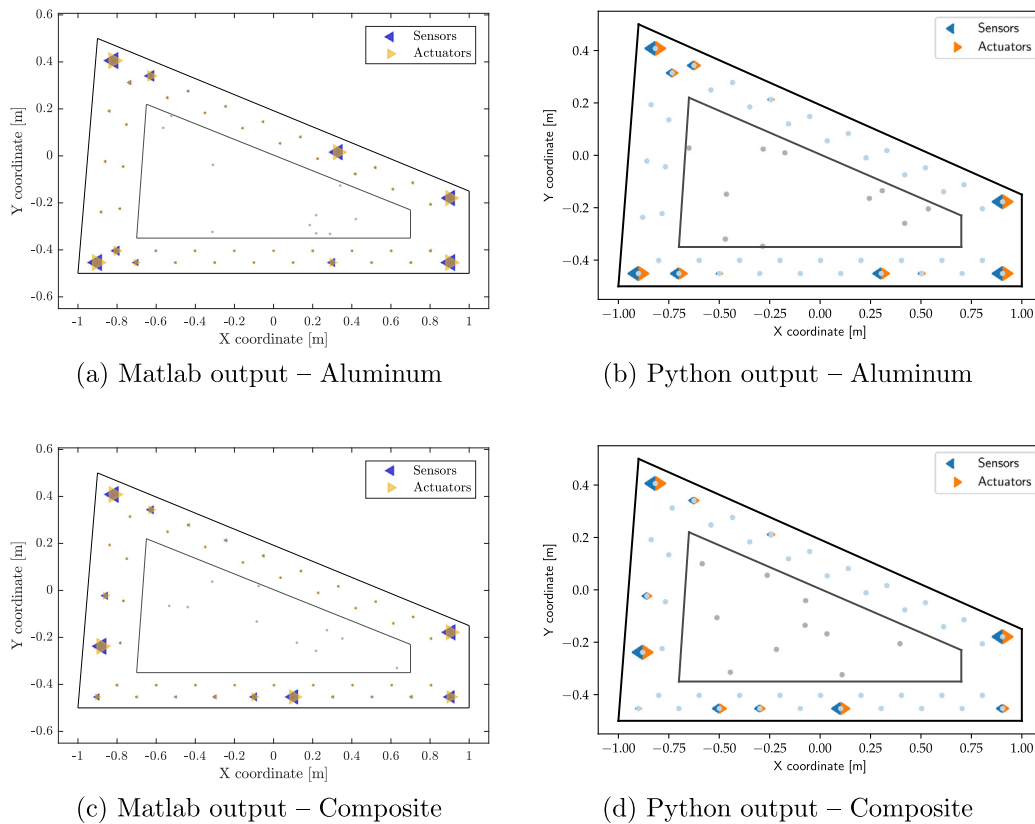


Fig. 5. Optimal configurations obtained with OptiSens in Matlab and Python implementations.

and (2) the minimum distance between two consecutive points in the space left between the outer plate geometry and the area of possible damage occurrence (refer to Fig. 3).

P_eval Creates the database of matrices P in Eq. (1) for each parameter configuration. For further details on the relation between Q and P matrices, please refer to Eq. (2) and [21].

objFun Evaluates both the objective function in Eq. (3) and its corresponding gradient. This gradient is used in the minimization algorithm to enhance its convergence.

More details on the specific input and output parameters for each functionality of OptiSens are provided within their implementations in Matlab and Python, which can be found in [27]. It is also worth mentioning that the optimization algorithms in Matlab and Python use different solvers to address its minimization. While Matlab uses the interior point algorithm [28], Python addresses the optimization using the sequential least square programming [24]. This difference might slightly affect the solution of the minimization problem, as shown in Section 4, but most of the optimal locations and the optimal number of sensors and actuators will remain the same.

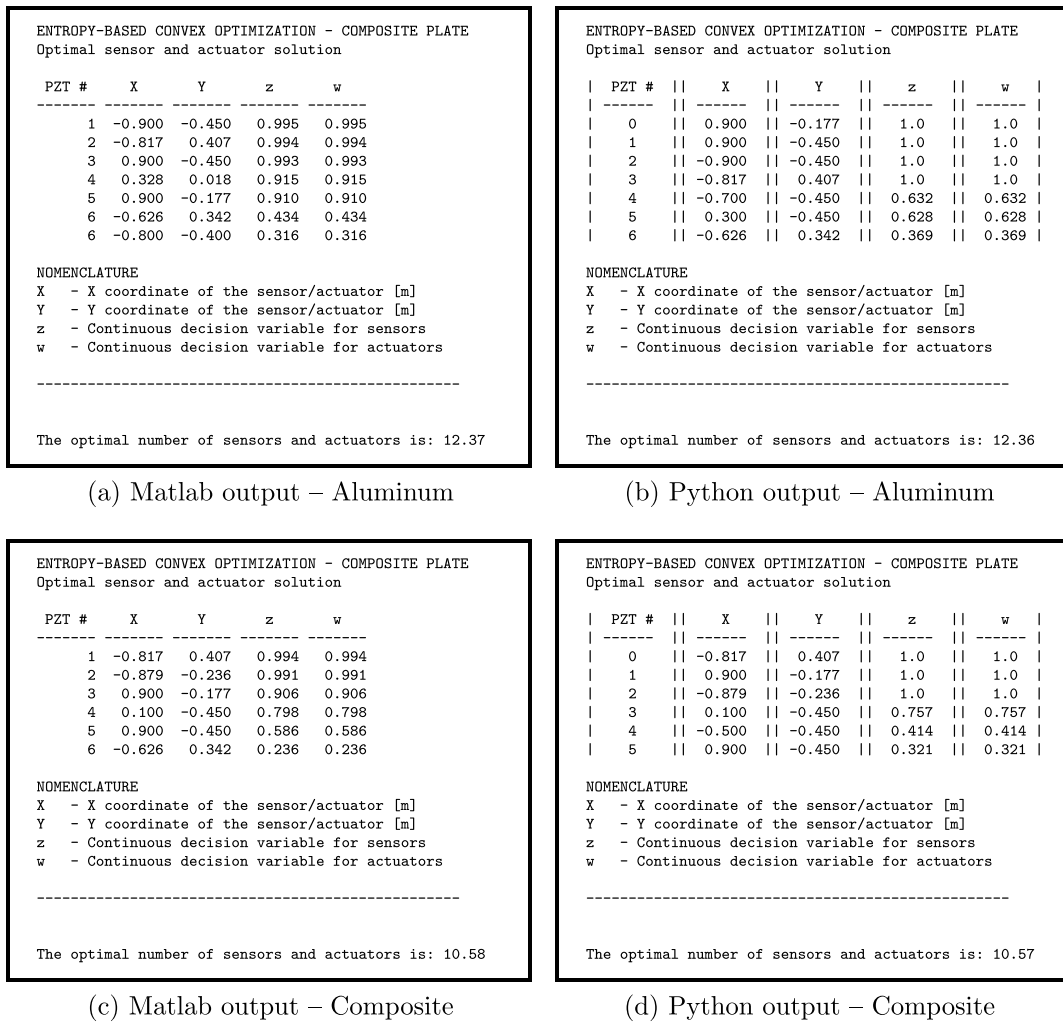


Fig. 6. Text files generated by OptiSens for each of its programming languages.

4. Illustrative examples

To demonstrate the capabilities of the proposed software, two case studies corresponding to equally shaped plates with equal areas of potential damage occurrence are provided. One plate is chosen to be made of aluminum, an isotropic material which has an approximate wave propagation velocity for the first anti-symmetric guided-wave mode of 2800 m/s at 300 kHz for a 1.4 mm thickness 2024-T351 aluminum alloy. The uncertainty related to this parameter is represented by a Gaussian distribution centered at 2800 m/s with standard deviation 60 m/s. The second plate corresponds to an angle-ply T800-M913 carbon fiber laminate of 3 mm thickness and stacking sequence $[-45_2/0_7/45_2/90_3]_s$, whose wave propagation velocity of the first symmetric guided-wave mode is different for different directions of space so it is characterized by two values modeled as Gaussian PDFs, namely $a \sim \mathcal{N}(6030, 40^2)$ and $b \sim \mathcal{N}(3549, 40^2)$, for the longitudinal and transverse velocities, respectively. The velocity values can be straightforwardly obtained using the Dispersion Calculator tool [29], whereby the dispersion curves of the guided-waves for different materials can be calculated. For demonstration purposes, the number of samples obtained from the prior PDF is set to 10. The dimensionless cost function $s(n)$ is constructed using a monotonic cubic-spline defined by the following interpolating points: (0,0), (30,0.3), (55,0.95), and (60,1) in pairs of $(n, s(n))$.

Fig. 5 shows the optimal sensor and actuator configurations obtained in both programming languages (Matlab and Python) along with the grid of possible sensor locations and the damage location samples used for the objective function evaluations. As evident from the figure, there are some differences between the optimal solutions provided by the two implementations. These differences stem mainly from: (1) the assumption of different samples randomly drawn from the prior distributions; and (2) the consideration of different solvers in the minimization of the objective function, as discussed in Section 3.3. Notwithstanding, the sensors and actuators rendering the highest values for the decision variables (i.e. z and w) result to be almost the same for both implementations.

Furthermore, the optimal configurations are also provided numerically in the text files, as shown in Fig. 6. These files show the most significant locations where the decision variables take the highest values using a table format, where the coordinates of these optimal locations are provided along with their respective decision variable values and the optimal number of sensors and actuators.

5. Impact

The OptiSens software platform has the potential to impact both the scientific and technological communities. Firstly, scientists working on novel modeling and optimization approaches

will be able to employ or modify the code in order to translate the approach into a different field (e.g., using a different sensing technology), or further improve the approach to produce results of higher performance. Moreover, from a technological perspective, OptiSens provides the first optimal sensor and actuator configuration tool for engineers working in ultrasonic guided-wave SHM system development. A noticeable example of the use of OptiSens is [21], where the configuration optimization of an ultrasonic guided-wave based SHM system for aerospace structural panels was addressed. SHM along with the wider field of non-destructive evaluation (NDE) are attracting increasing attention thanks to their potential to reduce maintenance costs and relaxing the currently strict design and certification rules. Research engineers may use OptiSens as a benchmark against which they can compare the efficiency and effectiveness of their own optimization codes, while practitioners will be able to easily learn how to determine the optimal number of actuators and sensors as well as their positioning, using a rigorous and efficient methodology. Furthermore, the SHM technology can be foreseen to receive increased attention in the near future and industrial leaders in the field [30,31] may wish to adopt OptiSens for consultancy projects or work closely with the authors to customize a proprietary version.

Overall, OptiSens is expected to impact the SHM and NDE fields. The determination of the influence area for each sensor and actuator has been historically dominated by simplified deterministic calculations, but we now have a rigorous tool which is computationally efficient and is also able to deal with real life problems while accounting for uncertainties.

6. Conclusions

OptiSens is a user-friendly and scientifically rigorous software platform for optimal sensor and actuator configuration for the health monitoring of either isotropic or composite plate-like structures. It relies on a cost-benefit analysis that numerically chooses the optimal number and position of sensors and actuators by minimizing the pre-posterior uncertainty of the damage coordinates and the wave propagation velocity.

Two different implementations are provided: a Python implementation whereby the input is directly introduced into the code by modifying some specific lines; and a Matlab implementation where the input is requested from the user by means of dialog boxes. Researchers or engineers who may want to modify or extend this software have a clean implementation in Python, while those who just need to use this tool may use the more user-friendly Matlab version.

Two case studies are provided to illustrate how OptiSens provides the optimal sensor and actuator configuration in a reproducible, effective, and understandable manner. Note that this software is available for downloading in GitHub [27].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This paper is part of the SAFE-FLY project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 721455. The authors are also grateful to the California Institute of Technology for kindly hosting the first author.

References

- [1] Giurgiutiu V. Structural health monitoring: with piezoelectric wafer active sensors. Elsevier; 2007.
- [2] Balageas D, Fritzen C-P, Güemes A. Structural health monitoring, vol. 90. John Wiley Sons; 2010.
- [3] Staszewski W, Boller C, Tomlinson GR. Health monitoring of aerospace structures: smart sensor technologies and signal processing. John Wiley & Sons; 2004.
- [4] Das S, Saha P, Patro SK. Vibration-based damage detection techniques used for health monitoring of structures: A review. *J Civil Struct Health Monit* 2016;6(3):477–507.
- [5] Goyal D, Pabla BS. The vibration monitoring methods and signal processing techniques for structural health monitoring: A review. *Arch Comput Methods Eng* 2016;23(4):585–94.
- [6] Kinet D, Mégret P, Goossens KW, Qiu L, Heider D, Caucheteur C. Fiber Bragg grating sensors toward structural health monitoring in composite materials: Challenges and solutions. *Sensors* 2014;14(4):7394–419.
- [7] Rose JL. Ultrasonic guided waves in solid media. Cambridge University Press; 2014.
- [8] Su Z, Ye L, Lu Y. Guided lamb waves for identification of damage in composite structures: A review. *J Sound Vib* 2006;295(3–5):753–80.
- [9] Aranguren G, Monje P, Kokonaj V, Barrera E, Ruiz M. Ultrasonic wave-based structural health monitoring embedded instrument. *Rev Sci Instrum* 2013;84(12):125106.
- [10] Cantero-Chinchilla S, Aranguren G, Malik M, Etxaniz J, Martín de la Escalera F. An empirical study on transmission beamforming for ultrasonic guided-wave based structural health monitoring. *Sensors* 2020;20(5):1445.
- [11] Papadimitriou C. Optimal sensor placement methodology for parametric identification of structural systems. *J Sound Vib* 2004;278(4–5):923–47.
- [12] Papadimitriou C, Lombaert G. The effect of prediction error correlation on optimal sensor placement in structural dynamics. *Mech Syst Signal Process* 2012;28:105–27.
- [13] Flynn EB, Todd MD. A Bayesian approach to optimal sensor placement for structural health monitoring with application to active sensing. *Mech Syst Signal Process* 2010;24(4):891–903.
- [14] Markmiller JF, Chang F-K. Sensor network optimization for a passive sensing impact detection technique. *Struct Health Monit* 2010;9(1):25–39.
- [15] Thieme M, Khodaei Z, Aliabadi M. Optimal sensor placement for maximum area coverage (MAC) for damage localization in composite structures. *Smart Mater Struct* 2016;25(9):095037.
- [16] Salmanpour M, Sharif Khodaei Z, Aliabadi M. Transducer placement optimisation scheme for a delay and sum damage detection algorithm. *Struct Control Health Monit* 2017;24(4):e1898.
- [17] Cantero-Chinchilla S, Chiachío J, Chiachío M, Chronopoulos D, Jones A. Optimal sensor configuration for ultrasonic guided-wave inspection based on value of information. *Mech Syst Signal Process* 2020;135:106377.
- [18] Li B, Wang D, Ni Y. On the high quality sensor placement for structural health monitoring. In: *Proc. IEEE INFOCOM*. Citeseer; 2009, p. 1–2.
- [19] Yi T-H, Li H-N, Gu M. Optimal sensor placement for structural health monitoring based on multiple optimization strategies. *Struct Des Tall Spec Build* 2011;20(7):881–900.
- [20] Ismail Z, Mustapha S, Fakhri MA, Tarhini H. Sensor placement optimization on complex and large metallic and composite structures. *Struct Health Monit* 2020;19(1):262–80.
- [21] Cantero-Chinchilla S, Beck JL, Chiachío M, Chiachío J, Chronopoulos D, Jones A. Optimal sensor and actuator placement for structural health monitoring via an efficient convex cost-benefit optimization. *Mech Syst Signal Process* 2020;144:106901.
- [22] Fritsch FN, Carlson RE. Monotone piecewise cubic interpolation. *SIAM J Numer Anal* 1980;17(2):238–46.
- [23] Ben-Tal A, Nemirovski A. Lectures on modern convex optimization: analysis, algorithms, and engineering applications, vol. 2. SIAM; 2001.
- [24] Kraft D. A software package for sequential quadratic programming. *Forsch be- Dtsch Forsch- und Versuchsanstalt Luft- Raumfahrt* 1988.
- [25] Cantero-Chinchilla S, Chiachío J, Chiachío M, Chronopoulos D, Jones A. A robust Bayesian methodology for damage localization in plate-like structures using ultrasonic guided-waves. *Mech Syst Signal Process* 2019;122:192–205.
- [26] Fendzi C, Mechbal N, Rebillat M, Guskov M, Coffignal G. A general Bayesian framework for ellipse-based and hyperbola-based damage localization in anisotropic composite plates. *J Intell Mater Syst Struct* 2016;27(3):350–74.
- [27] Cantero-Chinchilla S. OptiSens – Optimal sensor and actuator placement for ultrasonic guided-wave based SHM via an efficient convex cost-benefit analysis. In: *GitHub repository*. GitHub; 2020, <https://github.com/scaneroch/OptiSens>.
- [28] Waltz RA, Morales JL, Nocedal J, Orban D. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Math Program* 2006;107(3):391–408.
- [29] Huber A. Dispersion calculator user's Manual. Augsburg, Germany: German Aerospace Center (DLR); 2019.

- [30] Toullelan G, Chatillon S, Raillon R, Mahaut S, Lonné S, Bannouf S. Results of the 2016 UT modeling benchmark proposed by the French Atomic Energy Commission (CEA) obtained with models implemented in CIVA software. In: AIP conference proceedings, vol. 1806. AIP Publishing LLC; 2017, 150006.
- [31] Barras J, Lhémy A, Impériale A. Modal pencil method for the radiation of guided wave fields in finite isotropic plates validated by a transient spectral finite element method. *Ultrasonics* 2020;103:106078.